Exhibit 2
Accepted at IEEE Design & Test of Computing

4

A New Framework for Designing BIT MultiChip Modules with Pipelined Test Strategy

Ting-Ting Lin and Huoy-Yu Liou

Department of Electrical and Computer Engineering,
University of California at San Diego,
La Jolla, California 92093
lin@celece.ucsd.edu
liou@ece.ucsd.edu
(619) 534-4738
(619)534-2486 (fax)

Keywords: BIST, MCM Testing, Pipelining.

Abstract

In this paper, a novel test strategy, the Loop Testing Architecture (LTA) is introduced to reduce aliasing probability and testing time for multichip modules. This is accomplished by connecting Cascadable Built-In Testers (CBITs) in neighboring pipelined stages to increase the length of the test suites. Fundamental properties of LTA supporting the randomness in the generated test patterns (state coverage) and the asymptotic aliasing probability are presented. Our results on two small-scale multi-processor configurations show that the aliasing probability in analyzing signatures compared to that of a MLFSR [1] is comparable but with fairly low area overhead, and when compared with the Circular Self-Test Path technique [14], less testing time is required by LTA.

Further evaluation on the potential capabilities provided by the LTA, compared with boundary scan and other pipelined test scheduling approaches confirmed that LTA provides a new framework for designing effective testable systems.

Introduction

The next generation packaging, multichip modules (MCMs) which interconnect multiple bare dies by means of a stack of conductive and dielectric thin film, offers tremendous advantages such as reduced time delays between chips, less electrical noise and cross talk, simplified power distribution, and small size. However, large I/O lead counts and the high density interconnects decrease testing throughput and accelerate testing cost. Traditionally, testing is performed hierarchically. Chips are tested individually before assembly, and the assembled module is tested for correctness to avoid any errors introduced during packaging. Methods include bed-of-nails fixtures and hand held diagnostic probes become infeasible and cost-ineffective when new technologies such as MCM and surface-mounted devices are introduced. This is both due to often incomplete or unavailable test vectors from chip manufacturers and the internal module's low observability. The new approach in testing is built-in test (BIT), where a small circuitry is included in the circuit/system under test (CUT/SUT). Examples of well known BIT techniques are scan design [8], Built-in Logic Block Observer (BILBO) [9], etc.

Scan design methods involve disconnecting the memory elements and/or the flip-flops, from the combinational logics. The main problem with the Scan method is the overwhelming amount of test outputs generated by any relatively large circuit. One popular data compaction solution is signature analysis, which utilizes a linear-feedback shift register (LFSR) to receive and modify output data. The residue in the shift register, also called the signature, of a faulty circuit will differ from that of a good circuit after a long sequence of test patterns. Therefore, a combined boundary-scan and built-in self test (BIST) technique is recommended for board-level testing [8] to test complex circuits more efficiently.

BILBO is a technique that combines the basic features of scan design with those of signature analysis [9]. Feedback paths are formed in the shift registers by XORing some outputs from the flip-flops and connecting back to some of the inputs of the flip-flops. The XOR patterns is fixed for a given width of BILBO implementing primitive polynomial. Extra control ports are added to the shift registers where one combination of the control signals configures BILBO into a multiple-input shift register (MISR) for compacting circuit responses. The original 8-bit BILBO design has not kept pace with the bandwidth of today's computer design whose internal VLSI bus paths have long been extended from 8-bit to 16-bit or even 32-bit wide. Therefore, it is essential to redesign the BILBO to accommodate the wider bus path of today's complex VLSI systems. In [10], a family of concatenating polydividers with primitive characteristic polynomials were proposed trying to resolve the unextendable BILBO problem for packaged chips.

In order to minimize hardware overhead, the design time, but still maintain certain state and fault coverages, we propose a bytewise cascadable built-in tester (CBIT) macro cell with optimum primitive characteristic polynomial. The purpose is to keep the CBIT cell in a design library such that the circuit/system designers can easily construct the necessary feedback path for their BIST circuitry. Previous work on circular self-testing path (CSTP) [14] also accomplished cascadability, however, by simply connecting registers in a circuit to form a closed loop of which the feedback polynomial is $x^k + 1$. The choice of the feedback characteristic polynomial of the CSTP is non-primitive, therefore, the CSTP approach can be viewed as a special application of the CBIT. The performance of CSTP is not good as a result of its feedback polynomial being non-primitive. Specifically, sufficiently long testing time is recommended for CSTP's aliasing probability approximating to the asymptotic value, 2^{-N}, where N is the input width of the CUTs [14].

To further improve on testing time, we propose a novel approach, referred to as the Loop Testing Architecture (LTA), based on CBITs for testing MCMs concurrently. The LTA utilizes CBITs in a pipe interwoven with chips in high I/O count chips on MCMs. Simulation results show that this establishment guarantees high test coverage with the employment of maximum-length pseudo-random sequence (PRS) for test pattern generation. And the aliasing probability is comparable to that provided by a two-fold MLFSR [1] with only a fraction of the area necessary.

The need for a parallel exhaustive testing is significant for MCMs. The original test vectors for chips with high I/O counts from different manufacturers may not be available for the functional testing of the assembled module [15]. In this case, parallel pipelined exhaustive testing using LTA becomes imperative for the MCM designers to achieve better fault coverage in an efficient time frame than Boundary Scan. For chips without BIST circuitry, arrays of CBITs can be provided to the MCM in the forms of a small chip on the same substrate or off-MCM test circuitry. For chips with existing on-chip BIST structure, LTA can easily be supported.

This article discusses the CBIT design, the construction of the LTA pipes, and the underlying properties supporting such design. Measurable bounds for evaluation of the LTA from two sample systems will be presented, followed by comparisons with the existing boundary scan and pipelined BILBO *with conflict scheduling* [6] [7] in terms of testing time and area overhead.

Cascadable Built-in Testing Structure

The design goal of CBIT is to provide a macro cell in the design library expediting the BIT design process. CBIT cells are cascaded to form a *CBIT suite* utilizing multiplexors and XORs placed in strategic locations to construct different feedback paths, thus generating primitive polynomials in multiple byte configuration. A CBIT suite with feedback connections representing a primitive polynomial acts as a maximum-length PRS generator [5]. CBIT performs not only test pattern generation and signature analysis, but permits cascadability to generate a maximal length pseudo-random sequence (PRS). In performing signature analysis, a primitive characteristic polynomial gives a quicker convergence of the smaller asymptotic aliasing probability for a given test length [4].

The CBIT Design

A CBIT cell is a modified eight bit BILBO. It has three control signals [2]: C_x, C_y and C_z; eight parallel inputs (D-bus), eight parallel outputs (Q-bus), an LFSR consisting of eight flip-flops, and XORs providing feedback path of the LFSR. Two serial data ports, Scan_In and Scan_Out, are used for the scan path. Finally, Feedback_In and Feedback_Out provide the cascading links among CBITs. Fig. 1a shows the 8-bit CBIT cell and Fig. 1b is a 16-bit CBIT suite configured from two

CBIT cells [2].

The feedback pattern/generating polynomial for the CBITs is chosen so that the maximum-length PRS will be generated in both the 8-bit and 16-bit cases. Notice that in the 16-bit case, the feedback path for the least significant CBIT suite is different from the most significant CBIT suite (Fig. 1b) since the generating polynomial for the 16-bit CBIT has to be prime in order to guarantee the maximum randomness and quick convergence to the asymptotic value of the aliasing probability [4]. In general, CBITs can be cascaded to make extended length MISRs to fit the increasing size of the data buses without redesigning the detail of the BIST circuitry. This will help to speed up the design modification cycle to make the original designs more testable.

Operation Modes Provided by CBIT

There are three modes of operation (Fig. 2): parallel register, scan path, and MISR. The parallel register mode is for normal operation. CBITs form pipelined parallel registers in the data path. The scan path mode is for initialization and signature read-out. Non-zero seeds are shifted in via the Scan_In port and signatures are read out through the Scan_Out port. A scan path can be formed through the pipe to read out signatures in the intermediate stages as well.

For testing, the CBITs are configured in the MISR mode which concurrently perform pseudo-exhaustive test pattern generation for the succeeding CUT and output signature analysis for the previous CUT. The combinations of the three control signals, C_x, C_y and C_z, providing three major operations are summarized in Table 1:

Control Signals (C_x C_y C_z)	Configuration
(1 1 1)	parallel register mode
(0 1 -)	scan path mode
(1 - -)	MISR mode
(1 0 1)	most significant byte for cascading
(1 1 0)	least significant byte for cascading
(1 0 0)	single byte MISR

Table 1: Control signals and the corresponding settings of the CBIT operation modes

As shown by the last three rows in Table 1, the combinations of C_y and C_z enable the cascading of the CBITs.

Pipelining for Self Testing

Constructing A Pipe with CUTs and CBITs

In addition to the horizontal extension of the CBITs to accommodate large I/O MCM testing, further reduction in testing time can be accomplished when several functional blocks[1] in an MCM form a pipe where blocks in the pipe can be tested concurrently. Several pipes can be constructed according to their functionality and data widths. Each pipe consists of one zero-th stage CBIT suite, and subsequent stages of block and CBIT set.

Functional blocks with similar number of inputs/outputs are clustered to form a pipe. CBIT suites with corresponding width are then constructed to match the data width of each pipe. For those CUTs with very limited outputs (e.g., encoders), it is possible that more CUTs can be clustered and analyzed by the CBIT suites at each stage. Alternatively, several shorter or smaller width pipes can be constructed by the partition/segmentation process mentioned in [16]. Fig. 3a illustrates how pipes for a data path in the SUT are constructed, and Fig. 3b for a control path which usually has non-uniform input/output bit-width or branched signal flows. The proper length of any given pipe is determined by the requirements on the state coverage, the fault coverage, and the aliasing probability. Preliminary results can be found in our previous paper [2]. Once the set of pipes are formed, the number of stages in each pipe may be rearranged such that most of the pipes can finish self-testing simultaneously. Normally existing data paths with pipelining form natural self-testing pipes. When the pipe becomes too long that needs to be decomposed into two shorter ones, only the zero-th stage CBIT suite is added to the a second pipe. All pipes are created under this guideline after the rearrangement phase to give the maximum parallelism for this scheme.

For high fan-in CUTs, it is desirable to decompose the original network into segments with fewer fan-ins [16]. The controllability, detectability, and observability measures of a segmented circuit is exactly the same as that for the unsegmented CUT but with less computation effort [17]. Segments can be grouped into clusters by adopting algorithms proposed in, e.g., [18] and oftentimes, clusters identify natural LTA pipes

The Loop Testing Architecture (LTA)

Because of the degeneration of the cumulative test results over multiple stages, there exists the need for higher test coverage and lower aliasing probability in the pipelined MISR operation. One such improvement involves further cascading CBITs in neighboring stages using the Feedback_In/Feedback_Out lines to increase the length of the CBIT suite. The scan lines can also be constructed to facilitate scanning out signatures from all CBITs serially after the test session. This is referred to as the Loop Testing Architecture (LTA).

Fig. 3a also illustrates the construction of LTA where the Feedback_In is selected for the CBIT which performs the most significant unit analysis and Feedback_Out is selected for the least significant CBIT. The Scan_In and Scan_Out ports of the CBITs at each stage can be daisy-chained to give a scan path for the initialization and scanning out the final signature of each CBIT (also shown in Fig. 5). The last single CBIT suite of a pipe can be connected to the zero-th stage CBIT. Thus for each pipe, we have double-length CBIT suites for signature analysis which would result in smaller aliasing probability for the whole pipe. Fig. 4 shows the equivalent data flow when the CBITs are paired to do a double-length signature analysis. Those grayed functional blocks (F1, F2, F4, etc.) are replicated to show the paired testing flow when two CBITs are cascaded in the LTA.

[1] We refer the functional blocks to those CUTs in a SUT and modules to CUT/SUT with BIT circuits.

Arbitrary length of neighboring CBITs can be created using LTA for desirable aliasing probability.

Evaluation of the Pipelining Test

For testing, all of the CBITs are configured as MISRs, and are used for test pattern generation as well as signature analysis. This is justified by two assumptions [5]:

(i) The result of the input (seed) and the current state under the operation governed by the characteristic polynomial of the LFSR/MISR shall not be **0** for any state of the PRS.

(ii) Multiple inputs/seeds to the LFSR/MISR still traverse through all the states of the PRS; the degeneration/missing of some states in the PRS because of some special combinations/sequences of the seeds is not considered in current discussion.

These are further proved in [13] where the properties pertaining to the randomness of the patterns generated by a MISR exist even if the inputs are non-equal-probable.

To justify the pipelined LTA approach, we need to prove two things regarding the dual use of the intermediate CBITs. First, we need to show that these CBITs are effective as TPGs where patterns generated are indeed maximum-length PRS at each stage. This is supported by the pseudo-random property of the generating polynomial of the CBIT in the MISR mode, and is measured by the percentage of the corresponding maximum-length PRS. Second, we need to show that the limited output patterns of these functional blocks do not disturb the randomness of the signature, where the aliasing probability remains acceptably small after a number of stages.

Properties of the Pseudo-Random Test Pattern Generation from CBIT

When CBITs are constructed by LFSRs with primitive/irreducible characteristic polynomials, they possess the following major properties [5]:

(i) Every element/state α in the PRS generated by the LFSR has a complementary element/state $\overline{\alpha}$ in the same sequence such that $\alpha + \overline{\alpha} = \mathbf{0}$ (N-bit wide 0's), where '+' represents the operation on the two's complementary elements of the PRS defined by the characteristic polynomial of the LFSR.

(ii) For the cyclic PRS, more than one input seed will either decompose the original maximum-length cycle to more than one sub-cycles or merge at least two sub-cycles together.

(iii) The total number of (distinct) states of all the sub-cycles (if decomposed by multiple seeds) are $(2^N - 1)$ for N stage LFSR. In this manner, the **0 state** is excluded from the PRS and forms a trivial cycle ($\mathbf{0} \rightarrow \mathbf{0}$) for the LFSR.

When CBITs are cascaded to generate test patterns for different functional blocks in a pipe, we have the following observation:

(i) Each (at most) CUT with N-bit wide input buses only needs $(2^N - 1)$ different test patterns to finish the exhaustive self-testing.

(ii) In order to test the paired CUTs each with no more than N inputs exhaustively, at least $(2^N - 1)$ but no more than $(2^{2N} - 1)$ test patterns should be generated by one pair of cascaded CBITs. For example, given an eight-input CUT, we need $(2^8 - 1)$ test patterns. If there is no correlation between the two neighboring CUTs for one pair of cascaded CBIT suite, we need one maximum-length cycle of the 8-bit wide PRS to have one 8-bit CUT fully tested. However, because of the equally distributed 1's in the 16-bit wide PRS [5], two 8-bit CUTs should be fully tested before the extended PRS reaches its maximum-length period (which is $(2^{16} - 1)$). The actual number of the test patterns needed to fully test *m* CUTs simultaneously using *m* cascaded CBITs depends on the characteristic polynomial of the extended CBITs and the input seeds. But in general, we have the following relation:

$$(2^N - 1) \leq L \leq (2^{mN} - 1) \qquad (1)$$

where L is the test length needed to test *m* CUTs exhaustively given *m* CBIT suites cascaded for single stage analysis in a pipe.

Therefore, CBITs are effective as TPGs when the test length is appropriately chosen according to Eq. (1).

Aliasing Probability for Single Stage MISRs

CBITs in the MISR mode is a special case of the generalized linear-feedback shift register (GLFSR) [1]. A GLFSR(m, N) is a generalized m-stage, N-input signature analyzer with linear feedback patterns built over the Galois Field, GF(2^N). An N-input MISR is a case of GLFSR(m=1, N). Therefore, when the characteristic polynomials for the CBITs are designed to be prime, not only the test patterns are maximum-length but quick convergence to the asymptotic aliasing probability can be guaranteed.

Theorem 2 in [11] provides a general formula for calculating the aliasing probability for a one-stage, N-bit wide MISR:

$$P_{al}(p_0, p_1, p_2, \dots, p_{(2^N - 1)})$$

$$= 2^{-N}\left(\hat{p}_0^L + \sum_{i=1}^{2^N - 1} \prod_{j=0}^{L-1} \hat{p}_{i+j}\right) - p_0^L \qquad (2)$$

where L is the test length and $(\hat{p}_0, \hat{p}_1, \hat{p}_2, \dots, \hat{p}_{(2^N - 1)})$ are the Walsh transforms of *the error probabilities* from an N-bit output CUT. When $p_i = 0$, there is no error detection and $p_i = 1$, an error will always be detected.

Some closed forms of P_{al} exist with additional conditions [1]. Here we present two of the closed-form P_{al}'s by choosing the bit-error transition probability, p, to be 0.5 meaning that the probability of an output bit being erroneous is 0.5:

(i) When the test length L is $m(2^N - 1)$, where $m \geq 1$ is an integer, the aliasing probability P_{al} is

$$P_{al}(\frac{1}{2}) = 2^{-N} - 2^{(-mN)(2^N - 1)}, \text{ where } m \geq 1 \text{ is an integer.} \qquad (3)$$

for the *independent* error model [1].

(ii) When the number of test patterns is an arbitrary positive integer L and the probability of an output bit being wrong is 0.5, then P_{al} is

$$P_{al}(\frac{1}{2}) = 2^{-N}\left(1 - 2^{-NL} + (2^N - 1)\left(1 - \frac{2^{N-1}}{2^N - 1}\right)^L\right) \qquad (4)$$

for the 2^N*-ary symmetric-channel* error model [1].

Notice that for both cases when 2^N is much greater than one, P_{al} converges to an asymptotic value, 2^{-N}. Also, when the test length L is less than one maximum length for the N-bit wide MISR (i.e., $2^N - 1$) for the *independent* error model, Eq. (2) should be used for calculating the exact aliasing probability of the MISR.

Aliasing Probability in the Pipelining Scheme

In the multi-stage pipelining MCM testing scheme, the aliasing probability for the k-th stage can be calculated as

P_k (aliasing probability at k-th stage)
= 1 - (non-aliasing probability over k-stages)

$$= 1 - (1 - P_{al})^k$$

$$= 1 - \left(1 - k \times P_{al} + \frac{k \times (k-1)}{2} \times P_{al}^2 - \ldots\right)$$

$$= \sum_{i=1}^{k} (-1)^{i-1} \left(\frac{k!}{i!\,(k-i)!}\right) P_{al}^i$$

Let 2^N be much greater than one, which is generally true for all CBIT suites, P_{al} of the N-input MISRs can be approximated to 2^{-N} for all the values of m or L in the above formulae (2) and (3). Then P_k is simplified to

$$P_k \approx \sum_{i=1}^{k} (-1)^{i-1} \left(\frac{k!}{i!\,(k-i)!}\right) 2^{-Ni}, \quad \text{when } 2^N \gg 1 \text{ for pipes constructed by N-input MISRs.}$$

By ignoring the contribution from the higher power terms smaller than 2^{-N}, the aliasing probability for the *k*-th stage pipelined CBIT converges to

$$P_k \approx k \times 2^{-N} \tag{5}$$

Eq. (5) gives the asymptotic value for both the *symmetric-channel* error model with any test length and *independent* error model with test length at least one maximum length. When the number of stages, *k*, is much smaller than the maximum length of the PRS generated by the MISR (i.e., $2^N - 1$), the aliasing probability at the *k*-th stage MISR in the pipelining scheme is of the same order of magnitude as $O(2^{-N})$. This is also validated in the previous simulation result in [2], where the aliasing frequency/probability stays as a constant of $O(2^{-16})$ over 6 stages in the pipelining path.

Thus, the randomness of the signature is preserved in the case of limited number of multiple inputs, and the aliasing probability is sufficiently small given that k, the number of stages, is small compared with the maximum length of the PRS.

Other LTA Applications

Capability for Testing the Interconnects

The interconnects among MCMs can be tested with the pipelining scheme by integrating two sets of CBITs next to the I/O pins in each module. The first CBIT set operates in the MISR mode for both input validation before the signals reach the internal logics and TPG for the internal logic blocks. The second set of CBITs operates in the MISR mode for output from the internal logic circuitry and TPGs for the interconnection to the next module. Verifying interconnects among the MCMs are viewed as the simplified CUTs with compatible data paths. The whole system can be looked as many CUTs (including the interconnects) to be tested under the LTA scheme.

Fig. 5a shows one CBIT suite placed at the primary outputs (POs) of each CUT. The zero-th stage CBIT suite is added in order to generate the pseudo-random test pattern for the 1st CUT. Neighboring CBITs can be cascaded as shown previously in Fig. 3a to test the modular functionality of each CUT. However, this implementation cannot test the interconnections among the CUTs. In Fig. 5b, extra CBIT suite is inserted near the primary inputs (PIs) of each CUT. Therefore, we always have two CBIT suites testing either a functional block or an interconnect pattern between two CUTs. This implementation provides a general approach which can test any fault patterns in all permutations; e.g., multiple stuck-at faults, bridging or coupling, pattern sensitive faults, etc. The reason being that the N-bit wide interconnect network often realizes fewer than $(2^N - 1)$ different patterns for implementing signal links between any two CUTs. However, our N-bit wide CBIT suite can generate $(2^N - 1)$ different test patterns to exercise the N-bit wide interconnect exhaustively.

In scheduling the testing for the interconnects, no extra modes are needed nor timing conflicts exist. This is because we use two sets of the CBITs near the I/O pins which will transform the interconnects into another type of CUTs directly. Both the modules and interconnects can be tested concurrently in this pipelining scheme.

Area overhead resulting from ad ıpting the LTA for interconnection testing is caused by the insertion of one extra CBIT suite near the input ports, which makes the interconnects observable. Whereas only one CBIT suite at the outputs of each module is needed for performing the pipelining testing for the modules. By comparing the two schemes in Fig. 5, to test the interconnects and module functionality concurrently takes one more CBIT suite but saves separate mode(s) for reconfiguring the SUT to test the interconnects. Therefore, with a little area penalty, we can save a lot of testing time by testing the modules and interconnects simultaneously in one mode. Furthermore, the placement of CBIT sets still applies when the I/O ports are moved to the center of the modules.

Fault Location with Extra Observability of Intermediate CBITs

Signatures of the CUTs are read out when CBITs are configured in the scan path mode. Oftentimes, a wrong final signature indicates that faults exist in the test pipe. However, it is possible that faults from different CUTs in a pipe can cancel with each other resulting in a good signature at the last stage. Therefore, it is important we know the exact test length applied to each CBIT suite such that signatures of the intermediate stages can be made observable, thus facilitating fault location. In this manner, we can perform better diagnosis by locating faults in some specific CUTs.

Examples

We develop two experiments to demonstrate the effectiveness of the proposed Loop Testing Architecture. The first experiment involves testing a homogeneous processor environment consisting of SN74LS181/ALUs. The second experiment is about a heterogeneous MCM system with several types of components. Both of these systems are transformed into test pipes. Results in test coverage and aliasing probability are presented for discussion.

Six-stage ALU Pipes

Test Setup

Six ALUs form a pipe with 16-bit CBIT suites inserted between the ALUs. The 16-bit CBIT suite is chosen as the TPG for the 14-bit input ALU, SN74LS181. The 8-bit output of the SN74LS181 is fed into another CBIT suite configured for signature analysis. In this experiment, we developed two pipes based on the Loop Testing Architecture: one implements primitive characteristic polynomial between the looped CBIT pairs and the other directly connects the feedback lines without changing the feedback pattern of each CBIT suite. We also reconstructed the straight pipe from [2] for baseline comparison.

Randomness of the TPG

We measure the randomness of the test pattern generation process at each stage of the three pipes for various test lengths. The purpose is to evaluate the effectiveness of the CBITs as test pattern generator when operating in the MISR mode as well as the impact of pipe length on the test pattern generation. For an N-bit wide CBIT suite, the randomness measure is 100% if 2^N test patterns are generated. Fig. 6 shows the randomness measure for each stage of the three different pipes. In all three configurations, the randomness levels off after the first stage indicating that the length of the pipe does not affect the random pattern generation process.

Also in our previous observation, the required test length, L, for the two N-bit CUTs under LTA testing (in this case, $m = 2$ for Eq. (1)), should be smaller than $(2^{2N} - 1)$. This is validated by the simulation result in Fig. 6 that all the ALUs in every cascading stage of the pipe can be exhaustively tested when L is about four times the maximum length of the N-bit wide CBIT suite. That is, instead of $(2^{32} - 1)$ (or even $(2^{28} - 1)$) for the two ALUs, only 4×2^{16} test patterns is needed to give a 100% randomness for the two ALUs at each cascading stage.

The cascaded CBIT suite implementing the LTA outperforms the straight pipe in producing the best random patterns. And LTA with the primitive polynomial is better than that implementing the non-primitive polynomial. As we increase the test length, the CBIT suites eventually generate 16-bit wide maximum-length PRS. This validates our earlier presumption that multiple inputs to the PRS generators still produces the maximum-length PRS [5]. Regardless of how the 8-bit wide outputs are connected to the CBIT suite (the higher, lower, or even the middle byte), after a sufficient long test length, e.g., four times of the maximum length, 100% randomness of the 16-bit wide PRS can still be reached.

Aliasing Probability of Signature Analysis

Single stuck-at-0 fault is insisted at one output bit at the 1st stage ALU in all pipes as a way to introduce faults. Signatures of each ALU collected after certain amount of test patterns being applied to the pipes are compared with known good signatures. Aliasing occurs when the signature of the faulty pipe results in the same signature as the fault-free pipes.

We compare this aliasing probability at the last stage of the three pipes resulting from a stuck-at-0 fault at the least significant bit of the output of the 1st stage ALU. As shown in Fig. 7a, the aliasing probabilities of a 16-bit CBIT suite approaches $O(2^{-16})$ limit as test length increases in all three pipes. The straight pipe tends to get aliases early for shorter test lengths, while the CBIT suite implementing the LTA do not exhibit aliasing effects until after sufficiently long test (In this case, the aliasing in the primitive LTA pipe has aliasing at the 6-th stage after 100 tests.). However, when the test length is smaller than the length of one maximum-length PRS, the aliasing probability is more pronounced. Fig. 7b shows aliasing occurs at the later stages before it comes to the first stage. For the first stage, aliasing occurs after more than 1000 tests are applied. Whereas the 6-th stage has aliasing at test length less than 10. This implies that there needs a 'warm-up' period of the pipelined LTA in order to reduce the aliasing probability at each stage for small test lengths.

In general, the aliasing probabilities in the CBIT suites implementing the LTA are smaller than the straight pipe CBIT suites. When only one CBIT suite in the last stage is used for comparison, all stay at O(2^{-16}) (Fig. 7c). If the contents of the two CBIT suites are read as the complete signature, the aliasing probabilities for both pipes implementing the LTA are O(2^{-32}) in our single stuck fault simulation, whose value is negligible comparing to O(2^{-16}). This is due to the extended width of the CBIT suite of 2N, 2 x 16. It is interesting to see in Fig. 7a and Fig. 7c that the cascaded CBITs with non-primitive characteristic polynomials give the same asymptotic value of the aliasing probability as that of the primitive feedback polynomials discussed in [4].

Area Overhead and Testing Time

The area overhead for implementing the LTA consists of the extra wiring to cascade the CBIT suites with the additional XORs implementing the primitive generating polynomial. As mentioned before, no extra circuitry is needed comparing with the boundary scan when we construct the scan path with the cascaded CBITs. The testing time for the LTA pipes are the same as that of the straight pipe. However, with a little bit more wiring, the LTA pipes provide extra observability at each stage in the pipe and a much lower aliasing probability with the extended signatures.

Pipes with ALU, Caches and RAM (P pipe)

Test Setup

In the second experiment, an MCM consisting of one SN74LS181 ALU, one 8-bit RAM, and two 16x8 data caches, are put in a four-stage testing pipe. One 16-bit CBIT suite is placed at the inputs of the ALU to perform TPG. Four 8-bit CBIT cells are inserted between the CUTs. The test patterns from the 8-bit CBIT connecting to the inputs of the RAM are de-MUXed to test both Address and Data inputs exhaustively. This is referred to the straight P pipe [3]. A similar LTA pipe is constructed with extra connection between neighboring CBITs such that paired 8-bit wide CBITs can perform 16-bit signature analysis for two CUTs simultaneously (Fig. 4), also referred to as the 'cascaded P pipe'.

Randomness of the TPG

Fig. 8 shows the randomness measure of each stage with different test lengths for the straight P pipe. In Fig. 8a, 100% randomness is reached in the latter stage CBITs, when the input test length is greater than 2^8 for the 8-bit analysis. Also in Fig. 8b, the zero-th stage CBIT gives 100% randomness after the input test length is greater than four times the maximum-lengths for the 14-bit wide input bus to the ALU. This is consistent with the previous experiment that 'warming up' the zero-th stage CBIT can improve the quality of the TPG. For the P pipe with cascaded CBITs, Fig. 9 shows the behavior of the cascaded P pipe, which is very similar to that of the straight P pipe. This reassures us the earlier assumption that maximum-length PRS will be generated as the test length is at least four times the maximum-lengths for a given data width of the CBIT suites.

In Fig. 8a and Fig. 9a, only four times of the maximum-length PRS generated by one CBIT suite is needed to test those neighboring CUTs in the LTA (i.e., 4×2^8) as opposed to the that from a double-width CBIT suite (i.e., 2^{16}). There is a difference in Fig. 9 from Fig. 8 when the P-pipe uses cascading CBITs. The convergent rate for the cascaded CBITs is quicker than that of a straight pipe. This is again similar to the results of the previous ALU pipes.

Aliasing Probability of Signature Analysis

The single stuck-at-0 fault is injected to the least significant bit of the output from the ALU. Signatures from the faulty pipe are compared with those from a fault-free pipe to calculate the aliasing probability. The aliasing frequencies per injected fault of the final signatures in the last stage CBITs of the two P pipes are shown in Fig. 10a. The signatures are read byte-wise from each CBIT cell for aliasing frequency calculation. Complete signatures for the extended CBIT pairs are taken for analysis also. All aliasing frequencies per injected fault of the 8-bit wide signatures converge to the asymptotic value, which is 2^{-8} for the 8-bit case. However, the P pipe with cascaded CBITs gives a smaller byte-wise aliasing frequency than the straight P pipe when test length is smaller than one cycle of the maximum-length PRS, i.e., 2^8. Aliasing for the extended signatures does not occur until the test length reaches one maximum-length for the 16-bit signature analysis, i.e., 2^{16} or 65536.

Fig. 10b shows the aliasing frequencies for the intermediate stages in the P pipe with cascaded CBITs. The last stage still gives the worst analysis result as that of the ALU pipes. Therefore, 'warming up' the P pipes seems to improve the test quality. In addition, the aliasing from extended signature analysis (in this case, it is $O(2^{-16}) \sim 1.52 \times 10^{-5}$, for the complete 16-bit wide CBIT suite), is plotted for comparison.

In the above experiments, randomness of the TPG and the aliasing problem for multiple stage PSAs are analyzed. As the input test length traverses through the whole cycle of the maximum-length PRS provided by the MISRs, all states in the PRS will be visited at lease once. Thus we have 100% randomness of the maximum-length PRS during the TPG process. This is especially true from our experimental results for the downstream stages in one pipe. For well-partitioned pipelined testing path, the aliasing probability will be of the same order as that for the serial signature analyzer (SSA).

Area Overhead and Testing Time

To implement the LTA in the cascaded P pipe, only the wiring connecting those CBIT suites are needed comparing to the straight P pipe. In order to have a primitive polynomial for the extended CBIT suite, spare XOR gates are provided for most/least significant suite configuration. In terms of time needed in the testing phase, again, we only need two modes for initializing the CBITs and MISR mode analysis.

Performance Comparison with Other Approaches

LTA with CBITs is compared with other testing approaches in two aspects: testing time and area overhead. These approaches include the Boundary scan (JTAG standard) and a pipelined

BIST with conflict scheduling. Testing time is calculated by adding the set-up time ($T_{set\text{-}up}$), the module testing time (T_{module}) and the read out time ($T_{read\text{-}out}$) normalized to the average testing time per module. Area overhead is also calculated by adding hardware components necessary plus 40~80% extra for wiring.

The LTA does not need different CBIT cells in the design library to test different width of the data paths. For a wider data bus, we can cascade the CBITs to get an extended PSA without downgrading the quality of the signature analysis. Thus the hardware penalty required by different sizes of BILBOs can be eliminated. (The only hardware overhead needed by the LTA is the zero-th stage CBIT for a new pipe since the wiring for cascaded case is negligible.)

According to the previous work done on the performance analysis of CBITs, vs. boundary-scan [3], CBIT exhibits less than 10% of the testing time while requires only less than twice of the area than that of boundary-scan designs. In both cases, the fault-coverage is 100%. In our previous examples, it was shown that in the 6 stages of 16-bit pipelined CBIT suites, the aliasing frequency/ probability stays as low as $O(2^{-16})$ for a sufficient long test length. Therefore, with limited area penalty but an order of magnitude improvement of the total testing time, the LTA can drastically reduce the cost of MCM testing in today's competitive market.

Comparing with Boundary Scan

The boundary scan approach needs two separate modes and carefully selected test patterns to test the processor for the interconnect failure [8]. When the bit-width of the communicating data path gets higher, the more complicated test patterns and test cycles are required for the boundary scan.

Since the original test vectors may not be available for boundary scan test in the MCMs using ATPG [15], for an N-input CUT, the number of test patterns, *L*, needed for pseudo-random testing with boundary scan is $O(C \times 2^N)$ where $C \geq 1$ is a constant given by a statistical estimation on a specific test pattern generation technique [12]. The optimized value for C is one. However, to have certain test confidence that the most-difficult-to-detect faults are covered, a larger C is required [12]. The total time needed for one N-input CUT under boundary scan testing is

$$T_{set-up} + T_{module} + T_{read-out} = (C \times 2^N) \times (t_{set-up} + t_{module} + t_{read-out}), \qquad (6)$$

where $t_{set\text{-}up}$, t_{module}, and $t_{read\text{-}out}$ are the scan-in, one execution, and scan-out time for one CUT. But LTA gives

$$T_{set-up} + T_{module} + T_{read-out} = t_{set-up} + (4 \times 2^N) \times t_{module}/k + t_{read-out}, \qquad (7)$$

where 4×2^N is the maximum given by Eq. (1) for m=2 and k is the total number of stages of a pipe implementing LTA. Thus, LTA saves the scan-in/scan-out time and the time for pseudo-exhaustive testing per module/CUT, especially for $4 \ll k \ll 2^N$. For example, if $t_{module} = t_0$ clock cycles (t_0 is an equivalent number.), $t_{set-up} = 16$ clock cycles and $t_{read-out} = 16$ clock cycles for a 16-bit input, 16-bit output CUT, the boundary scan will need $2097152C + 65536t_0C$ clock cycles to finish the pseudo-exhaustive testing and LTA will take $32 + 32768t_0$ clock cycles when the number of stages, k, is 8.

Consider the interconnect testing, LTA does not require extra testing time in a separate mode that boundary scan [8] does. So, the total testing time per module and its interconnect will still be of $O(t_{set-up} + 2^N \times (t_{module} + (t_{interconnect} \sim 0) + t_{read-out}))$ since LTA can test both the interconnect and the processor logic at the same time. Here we write down $t_{interconnect}$ as the time for signals transferring through the interconnection network although its value is negligible comparing to other t's in the formula (For those interconnects with long propagation delay, $t_{interconnect}$ cannot be ignored.). For boundary scan, the total testing time per CUT with its interconnects will be in the order of

$$(C \times 2^N) \times (t_{set-up-all} + t_{module} + (t_{interconnect} \sim 0) + t_{read-out-all}),$$

where $t_{set\text{-}up\text{-}all}$ and $t_{read\text{-}out\text{-}all}$ are the sum of scan-in time and the sum of scan-out time needed for the CUT and its interconnect. The efficiency of LTA during the interconnect testing is again justified in saving more time than the boundary scan.

Considering the area overhead, both the LTA and the boundary scan approaches need five extra electrical pads for one processing element [8]. However, four I/O pins: test-mode-select, test-reset, scan-in, and scan-out are necessary for the control logic for the boundary scan[8].

In general, there is more area consumed by LTA with the XOR gates to implement the extended generating polynomial. Our previous experiment in testing the processor MCM in [3] shows LTA takes 4240 transistors in the four-stage straight P pipe case and boundary scan takes 3040 transistors in total. Therefore, LTA consumes about 39% more area than the boundary scan in this example. However, with limited area overhead, LTA provides a better BIT implementation with better state coverage and exponentially lower aliasing probability. Furthermore, LTA provides not only the extensibility for the PSAs in terms of the CBIT implementation, but also the pipelining for several CUTs to be tested concurrently to save the testing time per CUT.

Comparing with Pipelined BIST with Other Conflict Scheduling

Other pipelined BIST approaches in [7] alternates separate modes of TPG and PSA in one LFSR circuit. Implementation in [6] gives one stage analysis for all the CUTs in a pipelined data path by centralizing the TPG and distributing the PSA; i.e., one set of BILBOs as TPGs for all pipes with one stage BILBO-CUTs-BILBO structure and separating outputs to several PSAs. Conflict tables would have to be established to make sure the LFSRs perform TPG and PSA separately in different testing schedules. Here we denote them as the pipelining testing with conflict scheduling. The total averaged testing time for one N-input CUT ('kernel' from [7]) can be calculated as

$$T_{set-up} + T_{module} + T_{read-out}$$

$$= t_{set-up} + (t_{module} + (2^N - 1) \times D) + t_{read-out}, \tag{8}$$

where 2^N is used as the number of test patterns from [7] and D is the latency between the current TPG and its immediate predecessor (TPG). The minimum/optimized value for D is one clock cycle. Usually D is greater than one clock cycle since the new test pattern cannot be generated until the previous pattern is generated *and* loaded to the CUT/kernel when the bus is available.

By comparing Eq. (7) and Eq. (8), even though optimized pipeling schedule can be set for the best value of D, the conflict table requires more testing time then that of LTA. This is because our LTA approach utilizes the fundamental characteristics of the MISRs to operate simultaneously as TPG and PSA. Thus LTA eliminates the waiting time for the available register and bus to put a separate test pattern for the CUT/kernel. In addition, the possibility that more MISR circuit or interconnects required by the conflict scheduling for separating TPG and PSA modes does not occur in the LTA. For a 16-input and 16-output CUT, the testing time needed from Eq. (8) is at least $65567 + t_0$ clock cycles (again, $t_{module} = t_0$ clock cycles) which is greater than $32 + 32768t_0$ clock cycles given by the 8-stage LTA since t_0 should be at most one clock cycle.

Because of the dual TPG and PSA mode provided by the LTA, the exhaustive testing time is tremendously reduced by cutting the time for scheduling conflicts on one MISR (the D value of Eq. (8)). In addition, the extensibility given by LTA both horizontally for bit-size changes and vertically for multiple CUTs to be tested in a pipe provides best utilization of the parallel testing. Pipelining and parallelism can be performed on one system with minimized design modification and optimized test scheduling.

Without losing the effectiveness of the test coverage, the LTA scheme requires less testing time comparing to boundary scan and pipelining with conflict scheduling approaches. No significant area overhead comparing the former two implementations except the spare XOR paths for cascadability is anticipated in the LTA. Furthermore, we suggest that by re-arranging the placement of the CBIT circuits and test scheduling, it is possible we can gain high testability/ observability of the permanent faults both in the processor and in the interconnect.

Conclusion

In this paper, a cascadable Built-in tester (CBIT) is proposed to test MCM modules configured in a pipelined fashion. CBITs can be cascaded to match the data width of the CUTs and have been shown to exhibit high test coverage with 100% randomness in the TPG process and low aliasing probability in signature analysis. CBIT circuit can also serve as a switching device for module reconfiguration. This is achieved by the extra routing resources offered to the packaged MCM modules by current design houses. These routing layers can be used to reconfigure the interconnection when a die is diagnosed faulty.

We also introduced the Loop Testing Architecture as a way to reduce aliasing probability. When compared to the GLFSR approach [1], LTA gives similar aliasing probability as that of the two-fold GLFSR. LTA implementation can also be applied when the I/O ports are moved to the center of the chip area in the future system design.

LTA is more efficient when the MCM testing sessions can be partitioned into several sub-circuits for parallel testing. Partitioning algorithms using netlist as inputs can be found in [18] while partitioning at a higher level is proposed in [16]. In depth analysis of the testability issues for partitioned CUTs is discussed in [17]. Part of our future work will emphasize on integrating partitioning/clustering algorithms with LTA, such that hierarchical functional test methodology for MCM can be automated. Work on modification to the LTA and CBIT circuitry to accommodate the capability for interconnect reconfiguration and self purging in the MCM is also expected to improve fault tolerance.

Acknowledgment

We would like to thank Harry Chen of Cadence Design Systems Inc. for valuable comments in clarifying some ideas in this work.

Reference

[1] D.K. Pradhan and S.K. Gupta, "A New Framework for Designing and Analyzing BIST Techniques and Zero Aliasing Compression", IEEE Trans. on Computers, June 1991, Volume 40, Number 6, pp. 743-763.

[2] T.T. Lin and C. Kaseff, "Performance Evaluation of Cascadable Built-In Tester for Large I/O Multichip Modules", In Proceedings of the Fourth Annual IEEE International ASIC Conference, September 1991, pp9-3.1 -p9-3.4.

[3] T.T. Lin, J. Comito, and C. Kaseff, "Evaluation of Test Strategies for Multichip Modules", In Proceedings of the Fifth Annual IEEE International ASIC Conference, September 1992, pp. 234-237.

[4] M. Damiani, P. Olivo, M. Favalli, S. Ercolani and B. Ricco, "Aliasing in Signature Analysis Testing with Multiple Input Shift Registers", IEEE Trans. on CAD, vol. 9, no. 12, December 1990, pp. 1344-1353.

[5] Solomon Golomb, "Shift Register Sequences", Aegean Park Press, Laguna Hills, CA, 1982.

[6] A. Krasniewski and A. Albicki, "Automatic Design of Exhaustively Self-Testing Chips with BILBO Modules", Proc. Intl. Test Conf., 1985, pp. 362-371.

[7] M. S. Abadir and M. A. Breuer, "A Knowledge-Based System for Designing Testable VLSI Chips", IEEE Design & Test, August 1985, pp. 56-68.

[8] C. M. Maunder and R. E. Tulloss, "Testability on TAP", IEEE Spectrum, Feb. 1992, pp. 34-37.

[9] B. J. Koenemann, J. Mucha, and G. Zwiehoff, "Built-in logic block techniques", Digestion of Papers of the 1979 Test Conference, October 1979, pp. 37-41.

[10] D. K. Bhavsar, "Concatenable polydividers: bit-sliced LFSR chips for board self-test", Proc. of the 1985 Intl. Test Conference, pp. 88-93.

[11] M. J. Karpovsky, S. K. Gupta, and D. K. Pradhan, "Aliasing and Diagnosis Probability in MISR and STUMPS Using a General Error Model", Proc. of the 1991 Intl. Test Conference, pp. 828-839.

[12] D. K. Pradhan (editor), "Fault-Tolerant Computing - Theory and Techniques", Prentice-Hall, Englewood Cliffs, NJ, Chap. 1, 1986.

[13] K. Kim, D. S. Ha, and J. G. Tront, "On Using Signature Registers as pseudorandom Pattern

Generators in Built-in Self-testing", IEEE Trans. on CAD, vol. 7, no. 8, Aug. 1988, pp. 919-928.

[14] S. Pilarski, A. Kransniewski and T. Kameda, "Estimating Testing Effectiveness of the Circular Self-Test Path Technique", IEEE Trans. on CAD, vol. 11, no. 10, Oct. 1992, pp. 1301-1316.

[15] C. A. Pina, "Implementation of A MCM Brokerage Service", Proc. IEEE MCM Conf., March. 1993, pp. 46-51.

[16] R. Srinivasan, S. K. Gupta and M. A. Breuer, "An Efficient Partitioning Strategy for Pseudo-Exhaustive Testing", Proc. 30th IEEE Design Automation Conference, June, 1993, pp. 525-530.

[17] S. C. Seth and V. D. Agrawal, "A New Model for Computation of Probabilistic Testability in Combinational Circuits", Integration, the VLSI Journal, vol. 7, no. 1, April 1989, pp. 49-75.

[18] C.W.Yeh, C.K.Cheng, and T.T.Lin, "A probabilistic Multicommodity-Flow Solution to Circuit Clustering Problems" in the digest of technical papers of the IEEE/ACM International Conference on Computer-Aided Design, November 8-12, 1992, Santa Clara, CA, pp. 428-431.

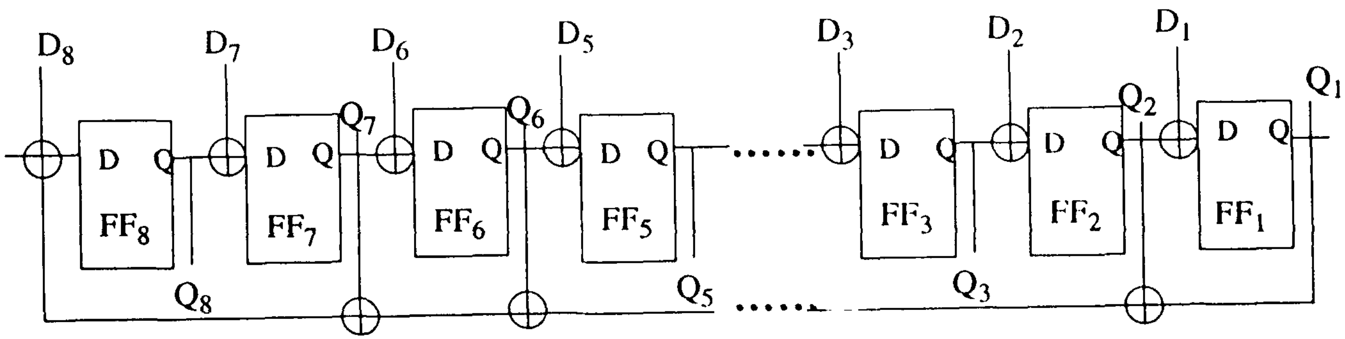


Fig. 1(a) 8-bit CBIT/MISR with generating polynomial = $x^8 + x^6 + x^5 + x + 1$

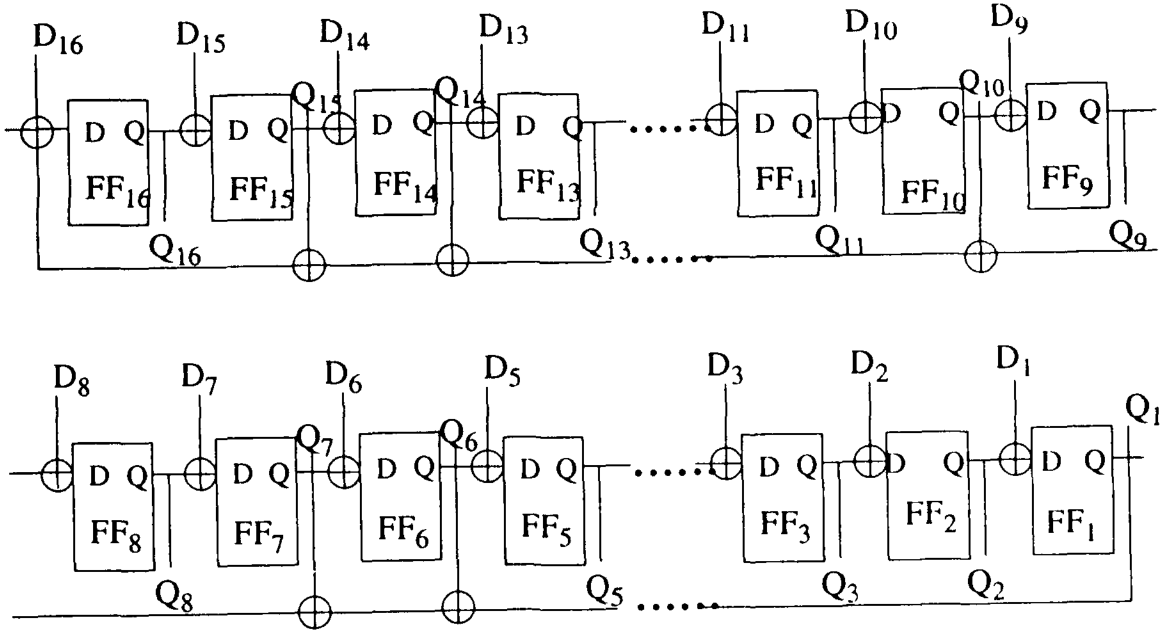


Fig. 1(b) 16-bit cascaded CBIT/MISR with generating polynomial = $x^{16} + x^{14} + x^{13} + x^9 + x^6 + x^5 + 1$

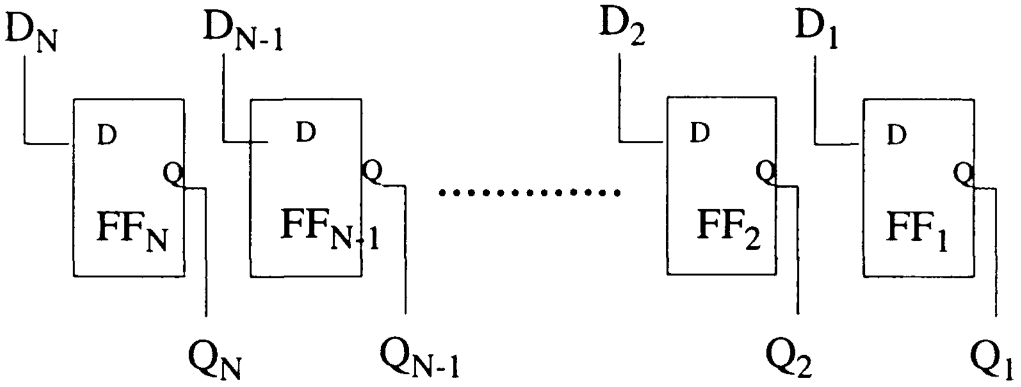


Fig. 2(a) parallel-in/parallel-out register/buffer mode

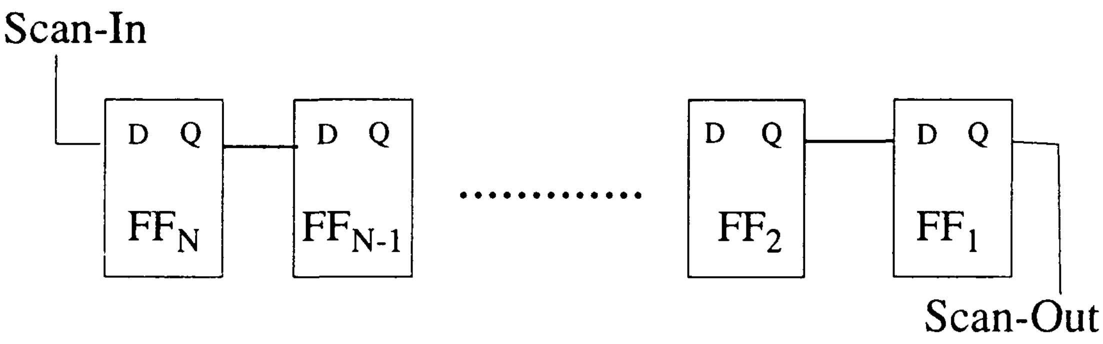


Fig. 2(b) Scan-in/Scan-out shift register mode

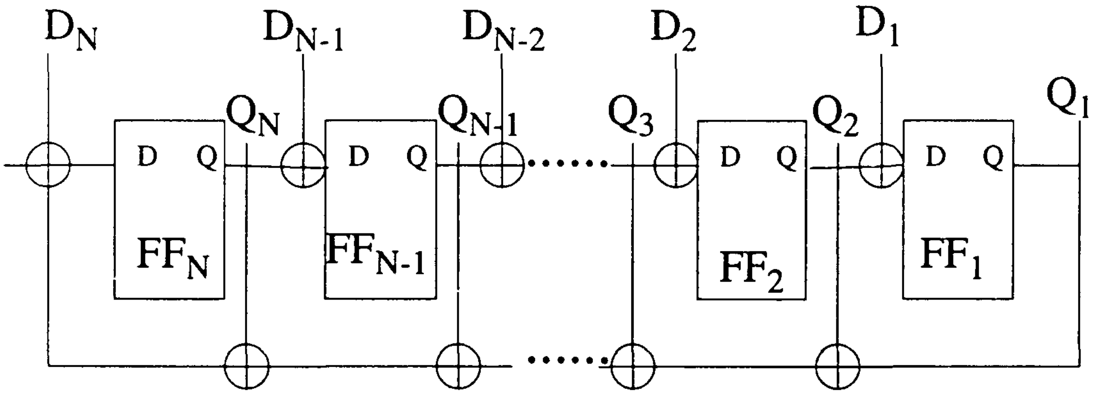
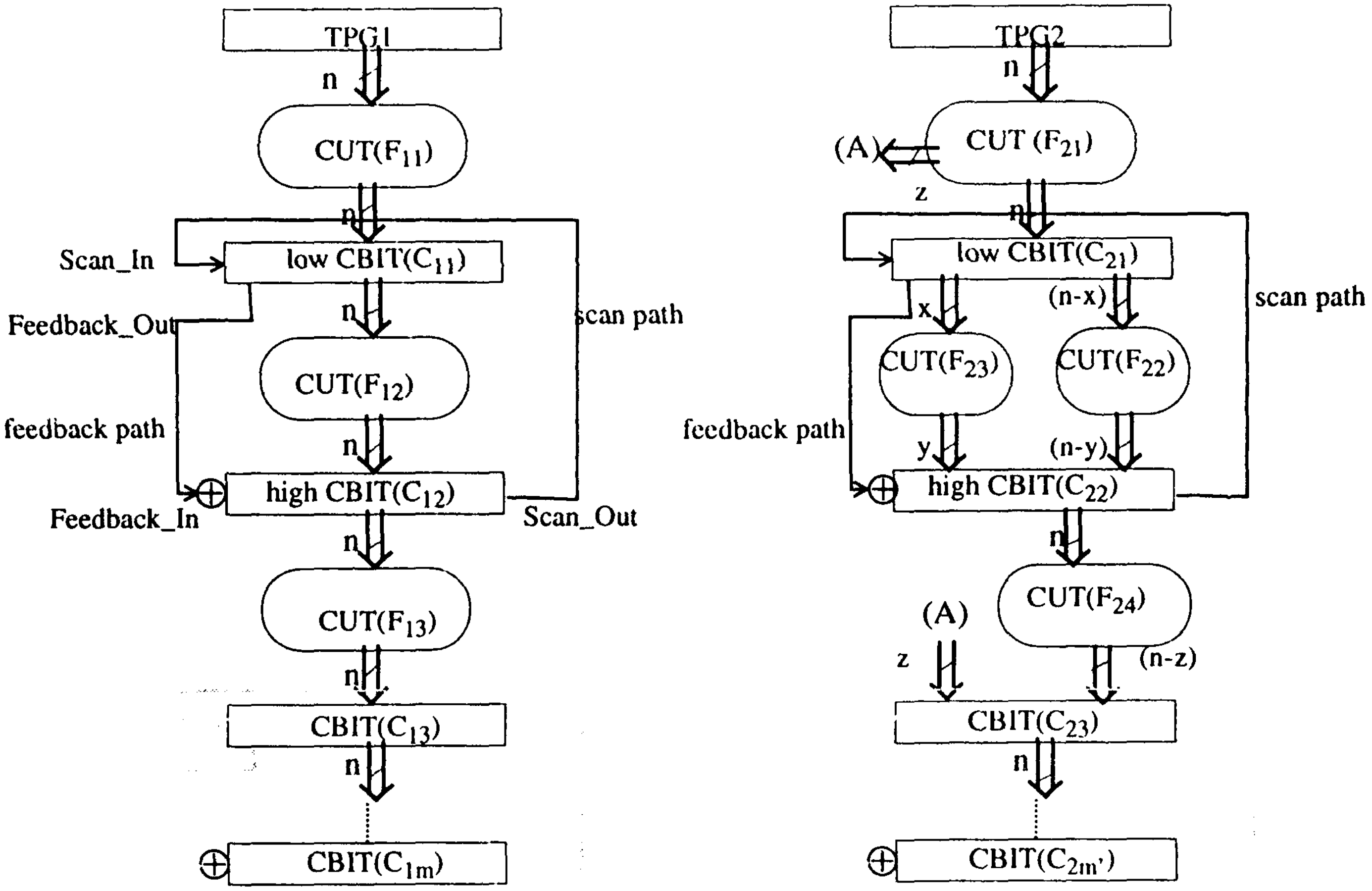


Fig. 2(c) parallel-in/parallel-out LFSR mode



(a) LTA pipe for CUTs with
homogenous data width
(mostly data paths)

(b) LTA pipe for CUTs with
heterogenous data width
(mostly control paths)

Fig. 3 Examples for constructing LTA pipes with paired CBITs in the system under test (SUT)

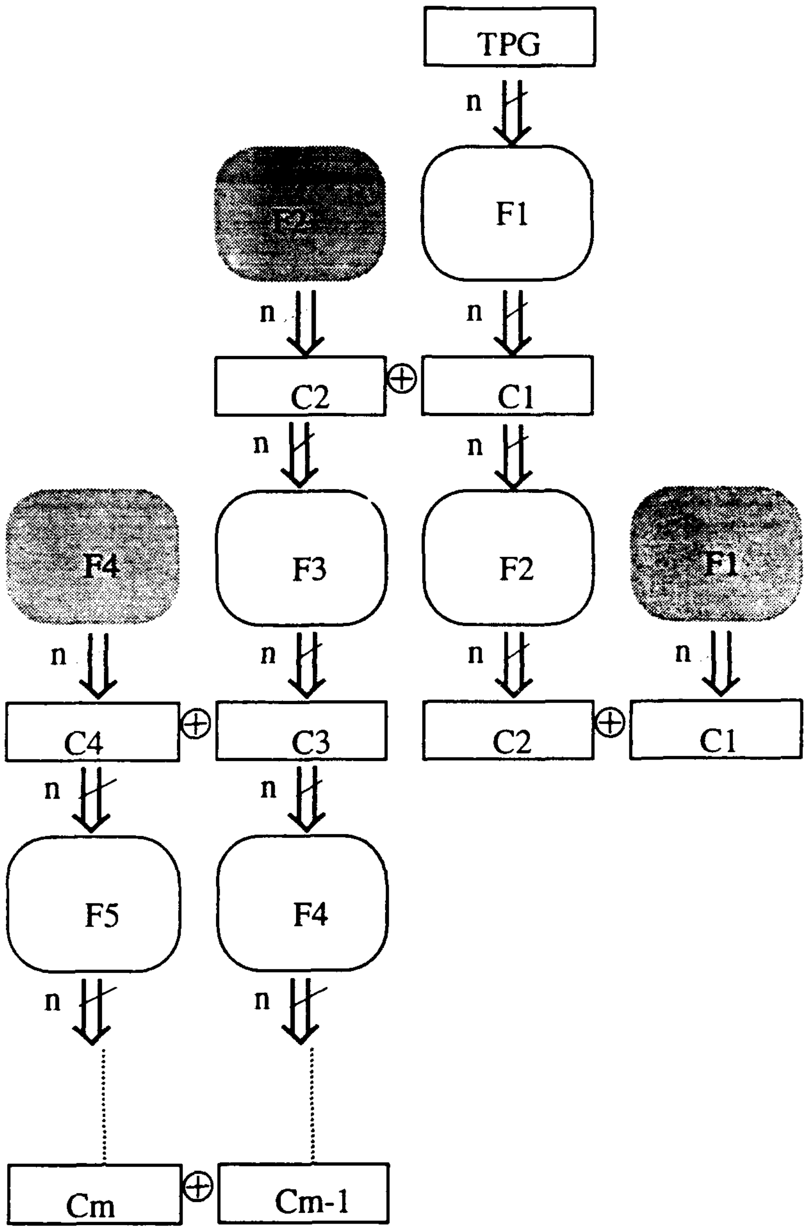
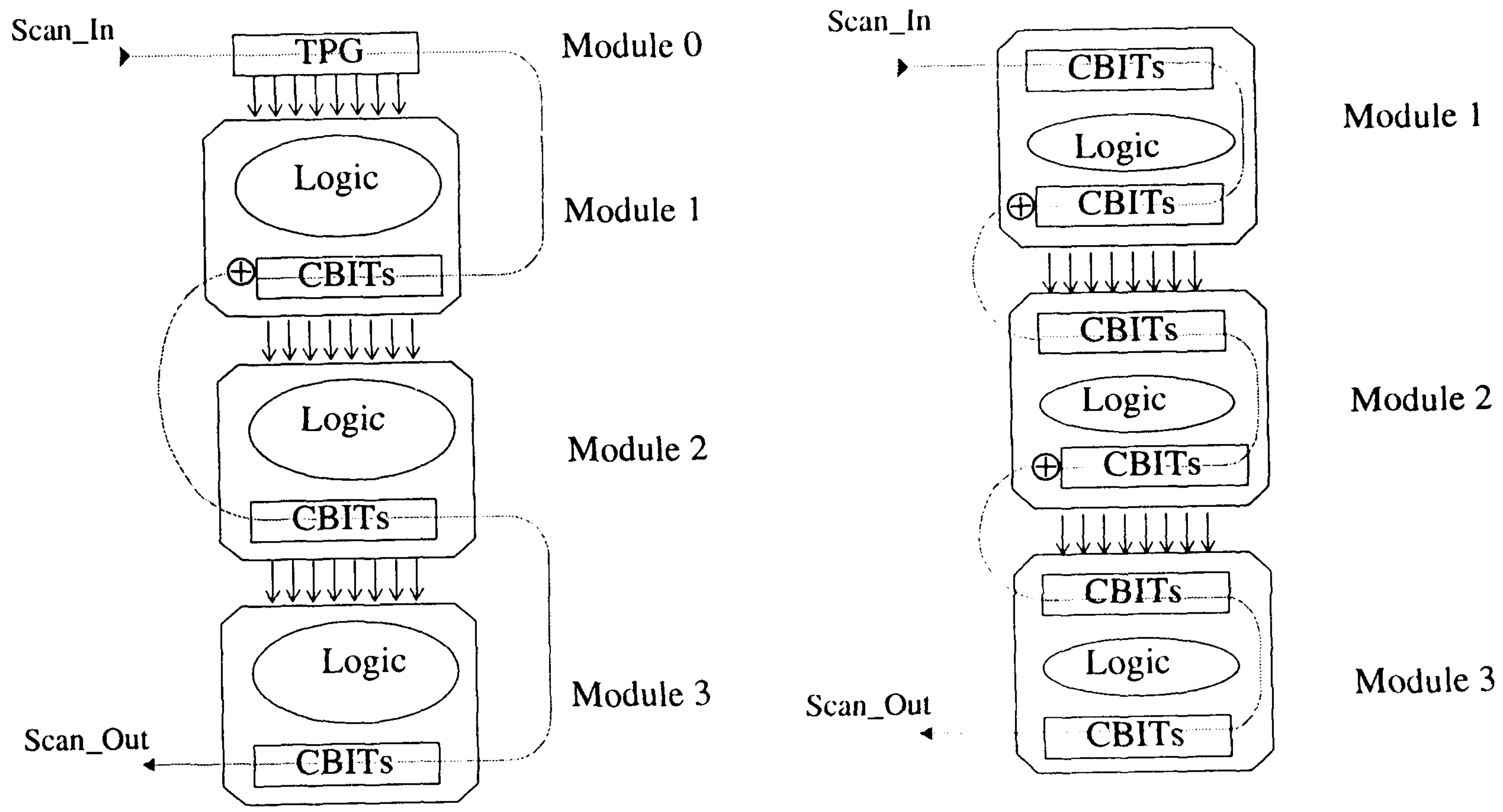


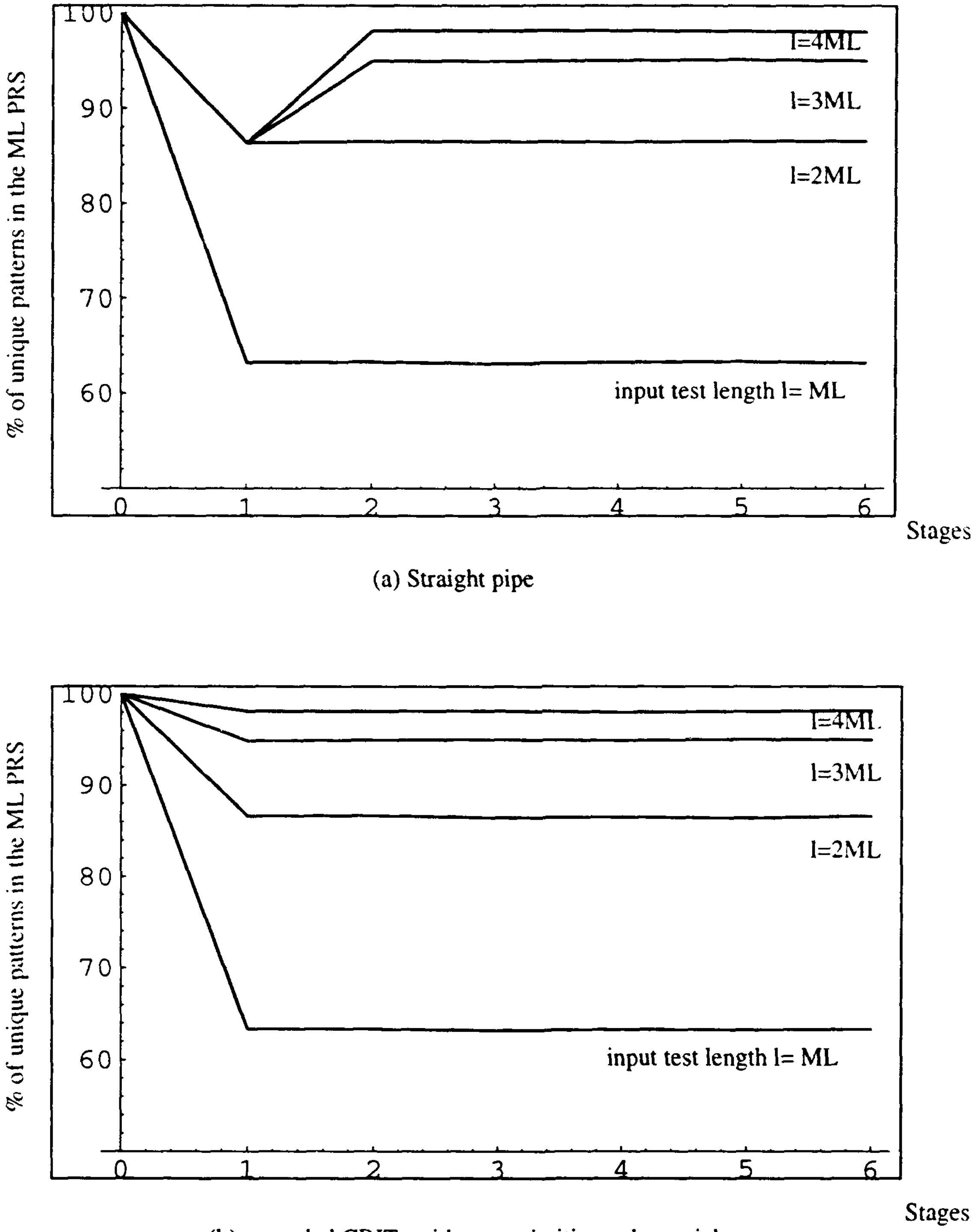
Fig. 4 Data path of the m-stage pipelined extended CBIT testing



(a) LTA for module functionality testing

(b) LTA for module functionality and interconnect testing

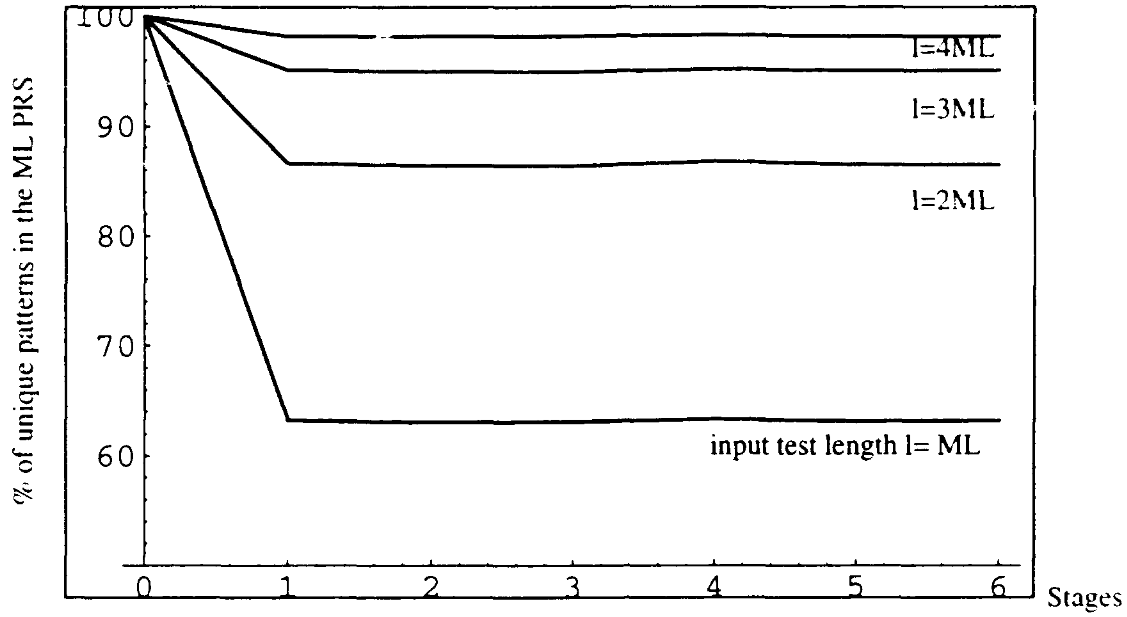
Fig. 5 The capability of LTA in testing the module functionality and/or the interconnects simultaneously (Sample scan paths with cascading connections are also constructed for both cases.)



(a) Straight pipe

(b) cascaded CBITs with non-primitive polynomial

Fig. 6 Randomness measure of the CBITs as TPG over 6 stages in the ALU pipes (ML=2^{16})



(c) cascaded CBITs with primitive polynomial

Fig. 6 Randomness measure of the CBITs as TPG over 6 stages in the ALU pipes (ML=2^{16})

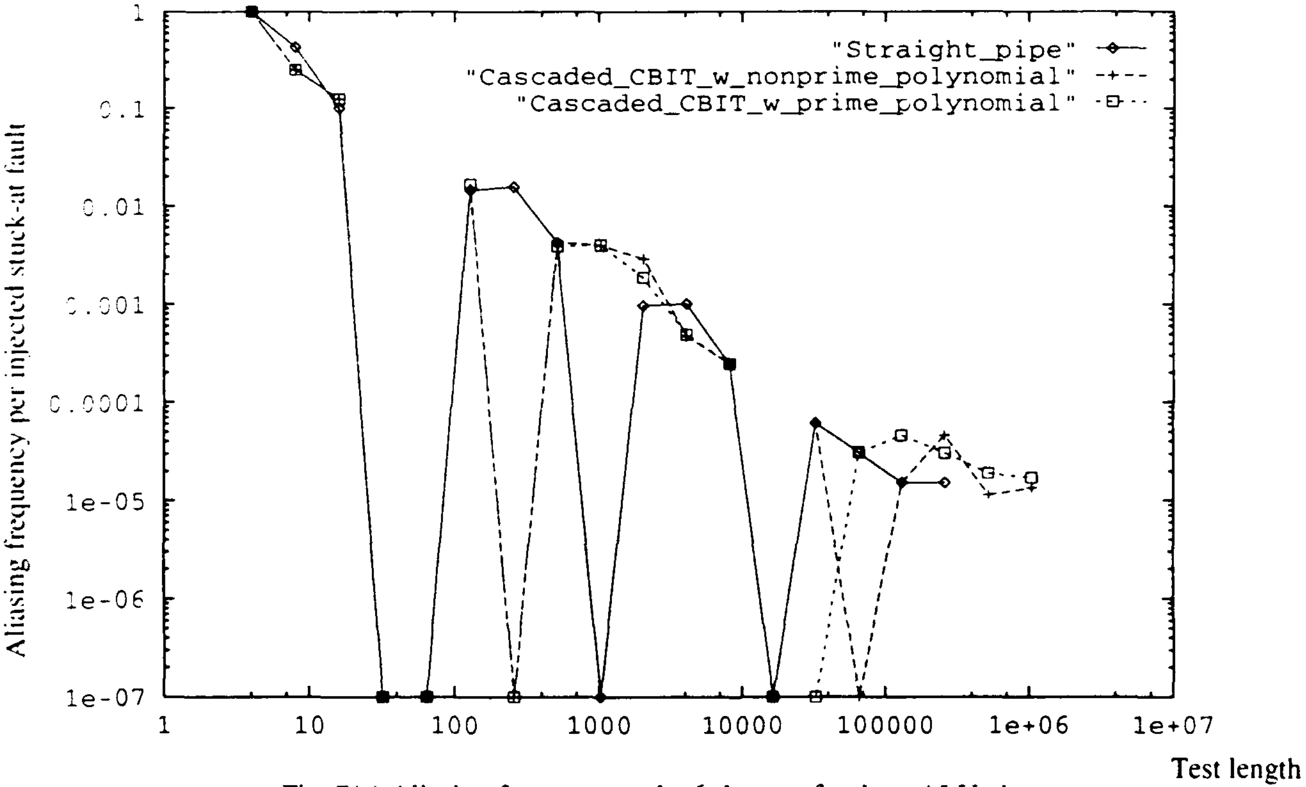


Fig. 7(a) Aliasing frequency at the 6-th stage for three ALU pipes

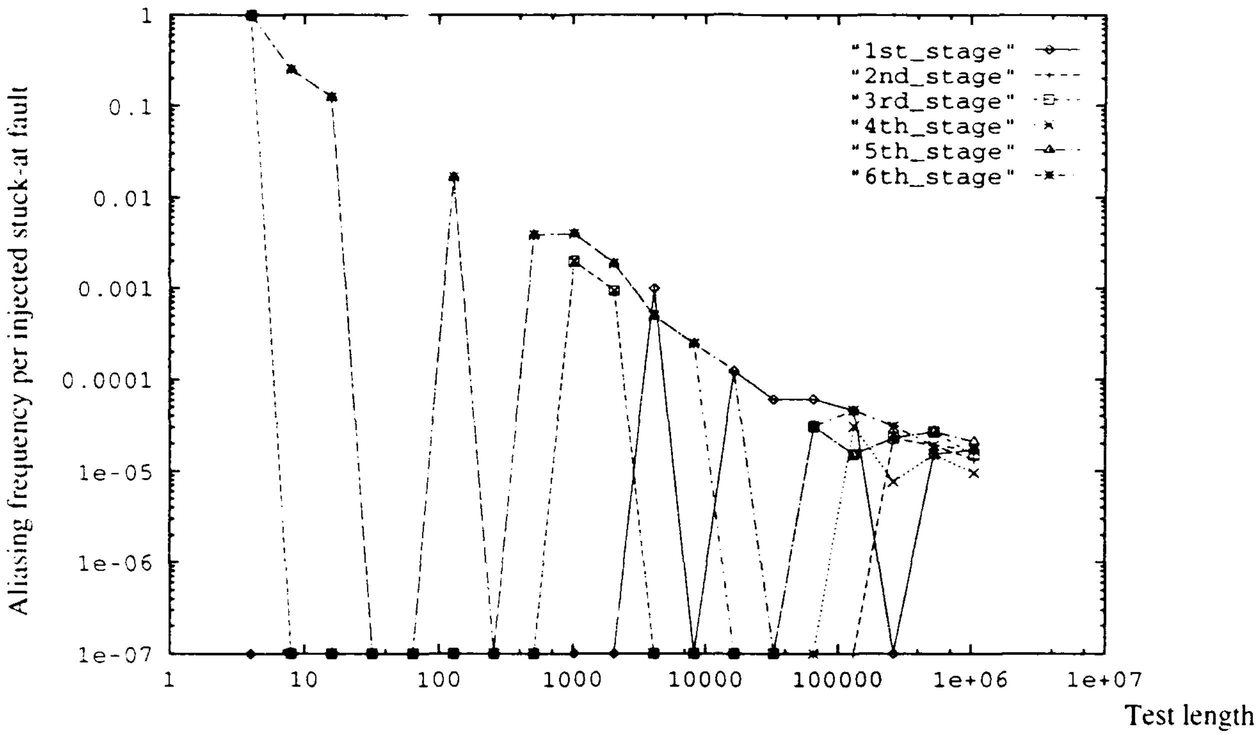


Fig. 7(b) Aliasing frequency for cascaded CBITs with primitive polynomial in the ALU pipe

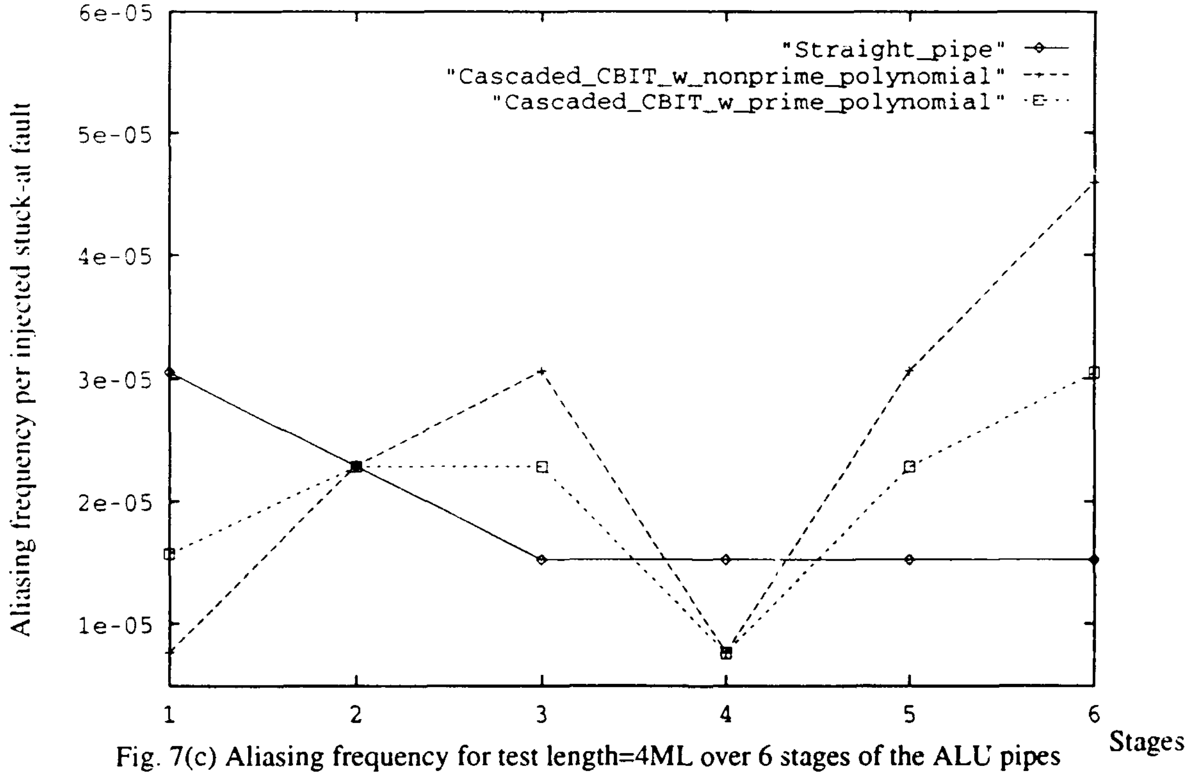
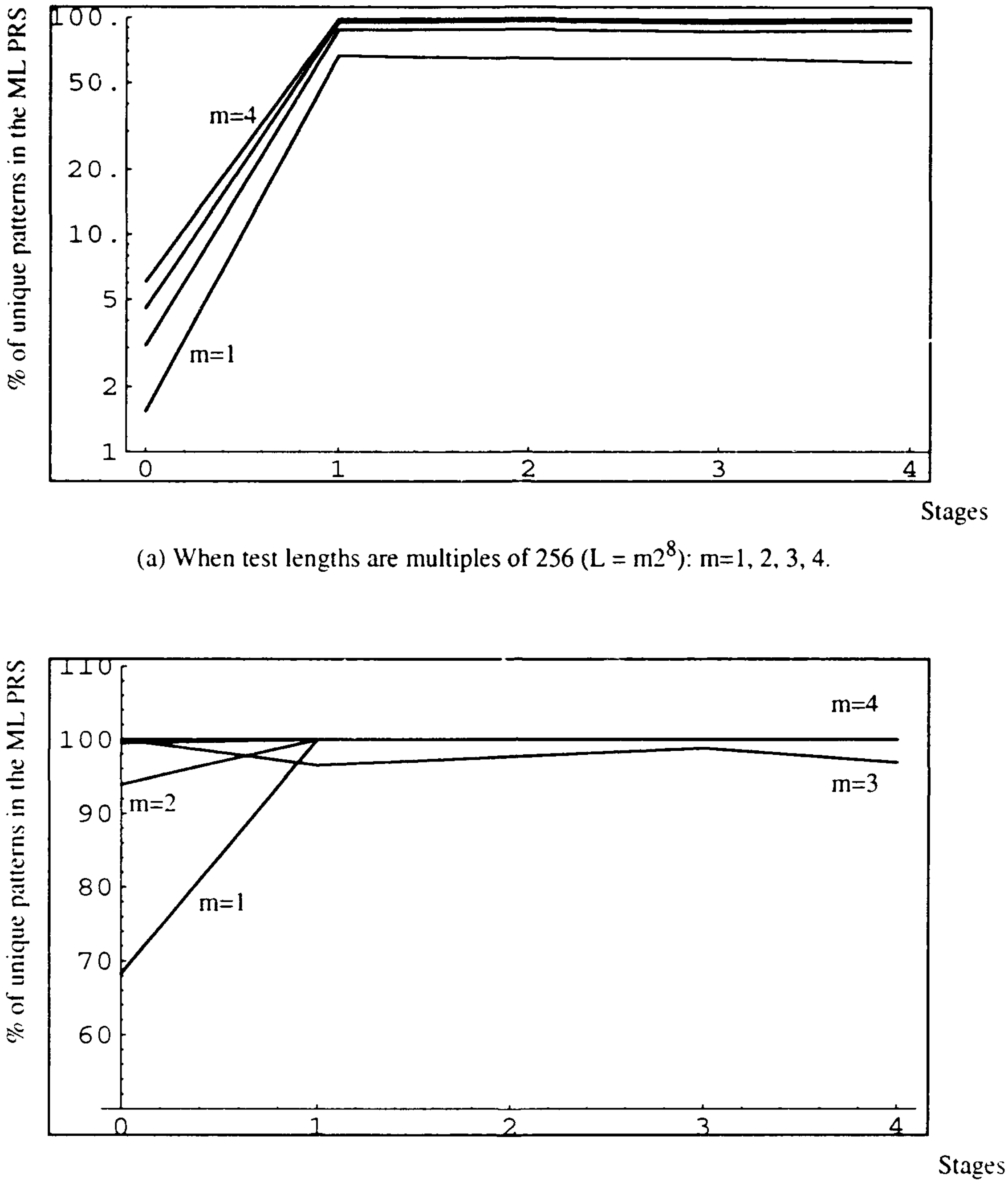


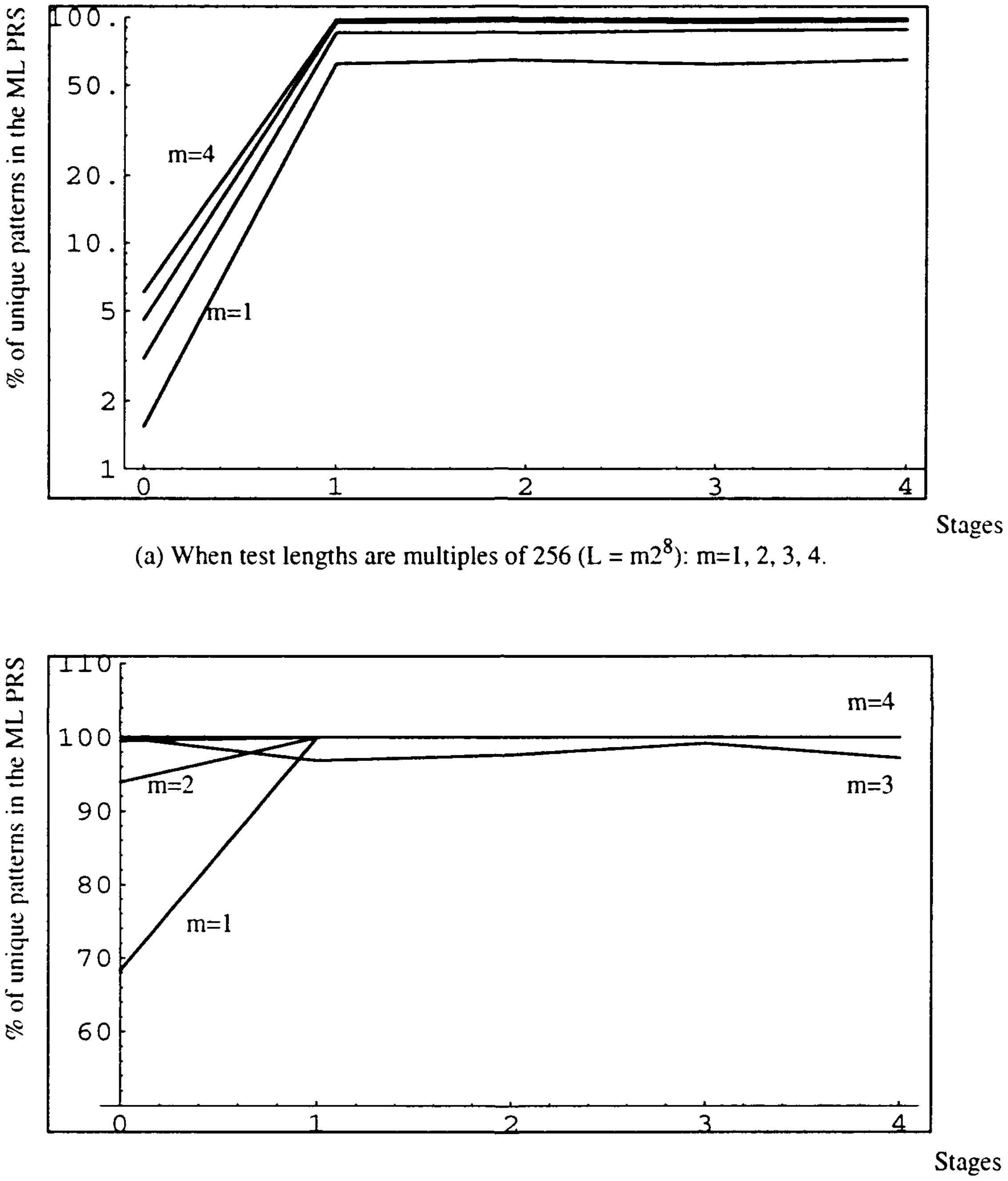
Fig. 7(c) Aliasing frequency for test length=4ML over 6 stages of the ALU pipes



(a) When test lengths are multiples of 256 ($L = m2^8$): m=1, 2, 3, 4.

(b) When test lengths are multiples of 16384 ($L = m2^{14}$): m=1, 2, 3, 4.

Fig. 8 Randomness measure of the CBITs as TPG over 4 stages for the straight P-pipe



(a) When test lengths are multiples of 256 ($L = m2^8$): m=1, 2, 3, 4.

(b) When test lengths are multiples of 16384 ($L = m2^{14}$): m=1, 2, 3, 4.

Fig. 9 Randomness measure of the CBITs as TPG over 4 stages for the cascaded P-pipe

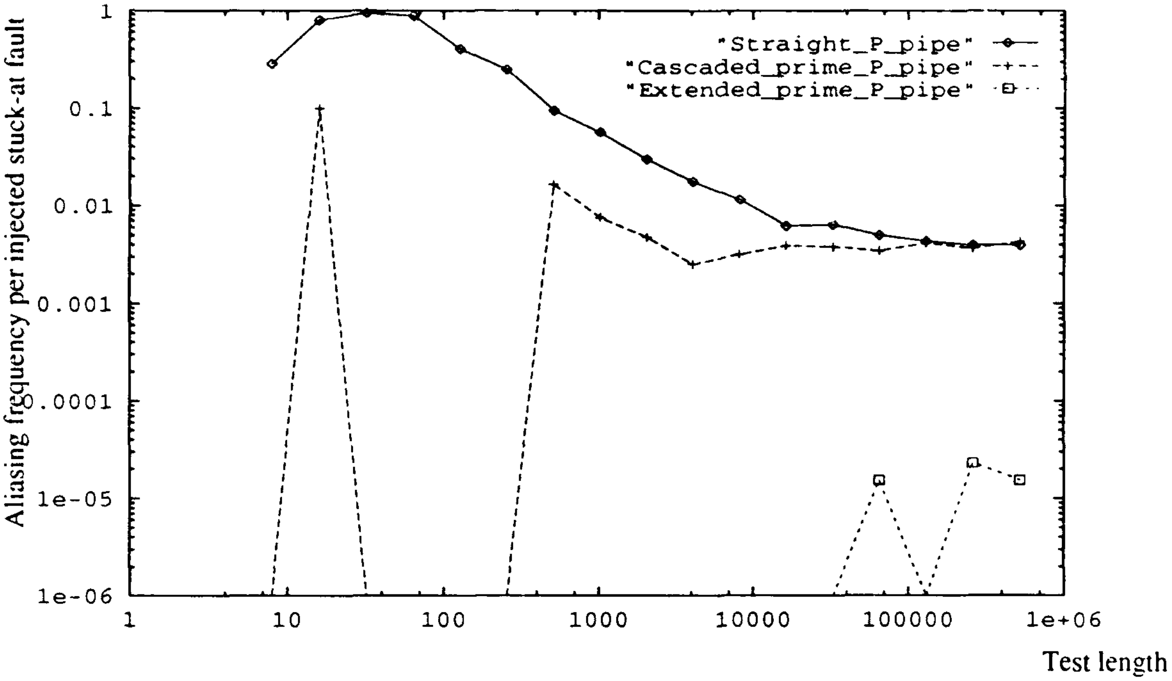


Fig. 10(a) Aliasing frequency at 4-th stage for two P pipes

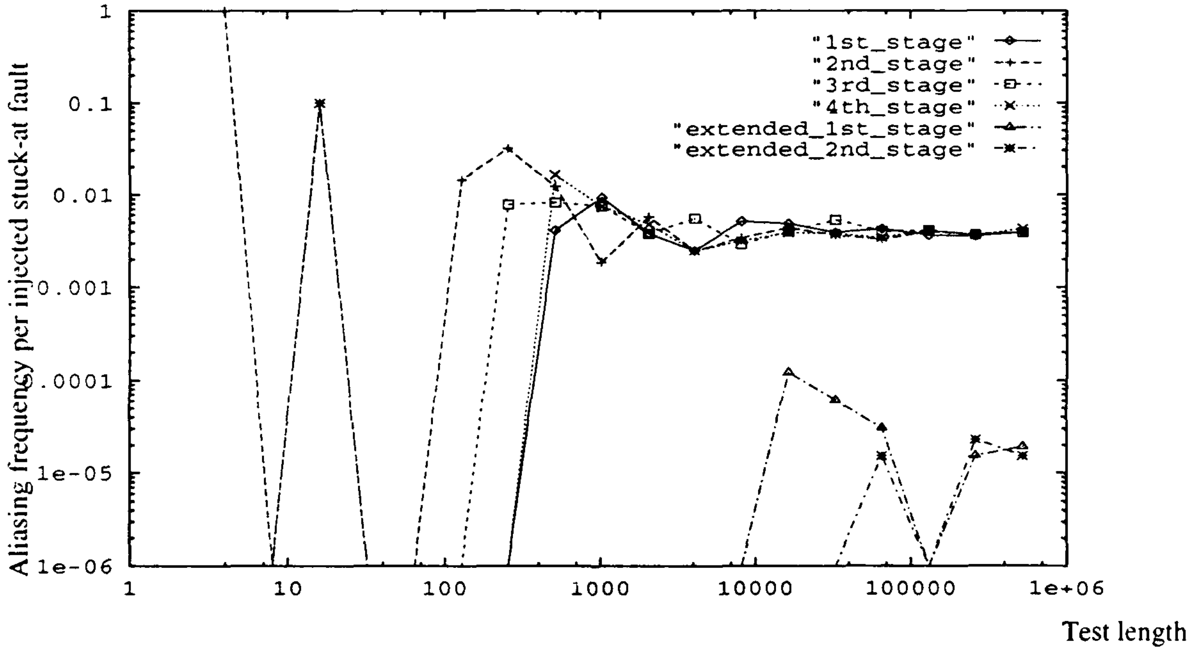


Fig. 10(b) Aliasing frequency for cascaded CBITs with primitive polynomial in the P pipe